N代表数组的第一维长度,M代表数组的第二维长度,N和M必须是常数或常量表达式,其值必须是正整数。这条语句相当于声明了 $a[0][0],a[0][1],\cdots,a[0][M-1],a[1][0],a[1][1],\cdots,a[1][M-1],\cdots,a[N-1][0],a[N-1][1],\cdots,a[N-1][M-1]这<math>N\times M$ 个元素,方括号内的数为元素下标,在程序中调用a[i][j]即可访问下标为i,j的元素。

BD 参考词条

基于指针的数组访问

🚣 延伸阅读

- [1] KERNIHAN B W, RITCHIE D M. C 程序设计语言[M]. 徐宝文, 李志, 译. 2 版. 北京: 机械工业出版社, 2004: 83-86, 95-97.
- [2] BJARNE S. C++程序设计语言(第1~3部分)[M]. 王刚, 杨巨峰, 译. 4版. 北京: 机械工业出版社, 2016; 150-152, 158-161.

(金靖)

1.2.7.2 数组的读入与输出

根据数组的定义,可使用循环结构来实现数组的读入与输出。

```
for (int i = 1; i <= n; i++)
    cin >> a[i]; // 输入 a[i]

for (int i = 1; i <= n; i++)
    cout << a[i] << " "; // 输出 a[i]
```

(金靖)

1.2.7.3 二维数组与多维数组

二维数组可看作一个N行M列的二维表格,也可以看作一个有N个元素的一维数组,而每个元素又都是一个长度为M的一维数组。在内存中存储时,二维数组按照先行后列的顺序连续存放数组元素,可以通过嵌套循环结构来访问其元素。超过三维的多维数组一般难以想象其对应的实际形态,但只需理解数组的每一个维度代表一类信息,常见于动态规划算法中记录多维状态信息。

BD 参考词条

- 1. 图的表示与存储: 邻接矩阵
- 2. 向量与矩阵的概念

(金靖)

1.2.8 字符串的处理

1.2.8.1 字符数组与相关函数

字符数组是一种 C 语言风格的字符串,在 C++中依然被保留下来。如下定义一个字符数组:

char a[10];

如果要在定义的时候进行初始化,可以用:

char a[10] = "Hello";

等价于

char a[] = {'H', 'e', 'l', 'l', 'o'};

其中,a[5] = '\0'。'\0'是字符数组的终止符,它的 ASCII 码是 0,有关字符数组的函数都需要利用终止符工作。对字符数组进行初始化或者读入等操作时,都会自动在字符数组末尾补上一个'\0',它也会占据数组的一个位置,如果没有预留这个位置就会造成数组越界。

除了数组本身的应用方式,字符数组还有其独有的函数。在 cstring 或 string. h 头文件中有许多对字符数组进行操作的函数,如表 1.3 所示。注意,使用这些函数前必须确保字符数组末尾有终止符'\0'。

函数	语法	功能
strepy	strcpy(a,b);	将字符数组 b 复制到 a 中, 并覆盖 a 中原有字符
streat	strcat(a,b);	将字符数组 b 拼接到 a 的后面
strlen	strlen(a);	返回 a 的终止符 '\0'前的字符个数
stremp	strcmp(a,b);	如果 a 和 b 完全相同,则返回 0 ; 否则设第一对不相同的字符为 $a[i]$ 和 $b[i]$,返回 $a[i]$ $-b[i]$ 的值(对应 ASCII 码相减)
strehr	strchr(a,x);	如果 x 在 α 中出现,则返回第一次出现的指针;否则返回空指针 $null$
strstr	strstr(a,b);	如果 b 是 a 的子串,则返回第一次出现的位置的左端点指针;否则返回空指针 $null$

表 1.3 字符数组函数

GD 参考词条

字符指针

₩ 延伸阅读

KERNIHAN B W, RITCHIE D M. C 程序设计语言[M]. 徐宝文, 李志, 译. 2 版. 北京: 机械工业出版社, 2004: 89-90, 145-146.

(金靖)

1.2.8.2 string 类与相关函数

string 是 C++的字符串类型, 需引用 string 头文件。如下定义一个字符串:

string a;

string a = "Hello"; //长度为 5,末尾没有字符数组终止符 '\0'

字符串的下标从0 开始。它支持用a[i]的形式访问下标为i的位置的字符,但如果超出当前长度就会造成越界。

字符串常用方法如表 1.4 所示。

方法	语法	功能
+	c=a+b;	字符串 a 的末尾添加字符串 b 并赋给字符串 c
+=	a+=b;	在字符串 a 的末尾添加字符串 b
push_back	a.push_back(x);	在字符串的末尾加人一个字符ェ
find	a.find(b,pos);	忽略字符串 a 中严格在 pos 之前的字符,如果字符串 b 是 a 的子串,则返回第一次出现的左端点的下标;否则返回 string::npos
insert	a.insert(pos,b);	在字符串 α 的下标 pos 之前插人字符串 b
size	a.size();	返回字符串 a 中的字符个数
substr	a.substr(pos,len);	返回字符串 a 中从 pos 开始的 len 个字符形成的字符串

表 1.4 字符串常用方法

参考词条

类的概念及简单应用

延 延伸阅读

BJARNE S. C++程序设计语言(第1~3部分)[M]. 王刚, 杨巨峰, 译. 4版. 北京: 机械工业出版社, 2016: 77-78.

(金靖)

1.2.9 函数与递归

1.2.9.1 函数定义与调用、形参与实参

如果一个程序中需要多次实现同一种功能,通常会将其设计成相对独立的代码段,

便于调用,这种代码段称为函数。函数是程序语言中的一种基本组成单位,通常以实现 某种功能为目的。有时也可以把复杂的功能分解成若干个相对简单的子功能,每个子功 能分别作为一个函数,用模块化的方式来设计程序,使得程序易于理解且便于维护。

函数由函数原型和函数体两部分组成,如下所示:

```
返回值类型 函数名(参数1类型 参数名1,参数2类型 参数名2,…){语句组 return 返回值;//类型须与第一行相同}
```

函数原型用于指定函数的名称、传入参数列表及返回值类型。函数的参数是可选的,可以有0至多个参数。每个参数都应写明类型。函数名称前面的返回值类型表示当函数执行完之后,返回何种类型的值。

函数体由{}中的一组语句组成,实现数据处理的功能。函数体中一般包含 return 语句,如果返回值类型为 void,则不用返回语句,或返回一个 void 类型的值。函数中可以有多个 return 语句,一旦遇到 return 语句,无论后面有没有代码,函数立即运行结束,将值返回。

函数的调用是指在一段程序中引用一个已经定义过的函数。一般来说,函数必须在被调用之前进行声明。例如,如果 main 函数中调用了 add 函数,add 函数的定义必须在 main 函数之前。但如果这样的自定义函数很多,那么 main 函数就会在程序代码中处于比较靠后的位置,这会影响阅读程序。为了避免这种情况,可以把函数原型声明放在函数调用之前,函数体放在函数调用之后。示例代码如下。

```
int add(int x, int y); //函数原型声明
int a = 3,b = 5;
int c = add(a,b); //声明整型变量 c 并赋值为函数调用返回值,即 3+5=8
int add(int x, int y) { //add 函数体
return x + y; //计算 x、y 的和,并将其结果返回给调用它的程序语句
}
```

程序语句调用函数时,将需要处理的数据作为参数传递给函数体。调用函数时传人的参数称为实际参数,简称"实参";在函数定义中出现的参数则称为形式参数,简称"形参"。在示例代码中,a 和 b 为实参,add 函数中的 x 和 y 为形参。

实参和形参在类型、数量和顺序上须保持一致。实参可以是任何表达式的求值结果,形参可看作占位符,在函数没有被调用时不会被赋值。当函数被调用时,形参被分配内存,接受实参的赋值,或作为实参的隐式引用。当函数调用结束后,立刻释放形参内存。所以形参不能在函数外部使用,其作用范围只在函数内部。

🔐 延伸阅读

BJARNE S. C++程序设计语言(第1~3部分)[M]. 王刚, 杨巨峰, 译. 4版. 北京:

1.2.9.2 传值参数与传引用参数

实参和形参用于程序语句和函数之间的数据传输。发生函数调用时,实参的值会传递给形参。传递分为值传递和引用传递两种。

值传递是一种单向的数据传送,只能把实参值的一份复制传送给形参,而不能把形 参的值反向地传送给实参。

```
void calc(int b) {
    b *= b; //语句执行完成, b 的值为 4
}
int a = 2;
calc(a); //语句执行完成, a 的值为 2
//实参的值不随形参的值的改变而改变
```

引用传递,是一种双向的数据传送,形参在函数内的变化也会导致实参的值发生变化。一个引用传递的形参需要在变量名之前加上 & ,即变量类型 & 变量名,例如 int &a。

```
void calc(int &b) {
    b *= b; //语句执行完成,b 的值为 4
}
int a = 2;
calc(a); //语句执行完成,a 的值为 4
//实参的值随形参的值的改变而改变
```

❷ 参考词条

指针

₩ 延伸阅读

BJARNE S. C++程序设计语言(第1~3部分)[M]. 王刚, 杨巨峰, 译. 4版. 北京: 机械工业出版社, 2016: 273-281.

(金靖)

1.2.9.3 常量与变量的作用范围

局部变量定义在函数内部,作用域仅限于函数内部。形参变量、在函数体内定义的变量都是局部变量。局部变量可以在不同的函数中使用相同的变量名,但它们的作用域不同,互不干扰。在语句块中也可定义变量,它的作用域只限于当前语句块。例如在for 语句块中定义的变量,当语句块结束后,非静态变量就不存在了。而 static 关键字定义的静态变量存储于进程的全局数据区,它的值不随函数的调用或返回而改变。

全局变量定义在所有函数的外部,它的作用域默认是整个程序,也就是所有的源文